CTL
Communications
Technology Laboratory

SST/SysML2 Semantic Assets and Debt : Associations / Connections

Conrad Bock
U.S. National Institute of Standards and Technology
Raphael Barbau
University of Maryland, NIST Associate
Ed Seidewitz
Model Driven Solutions

NIST
National Institute of Standards and Technology

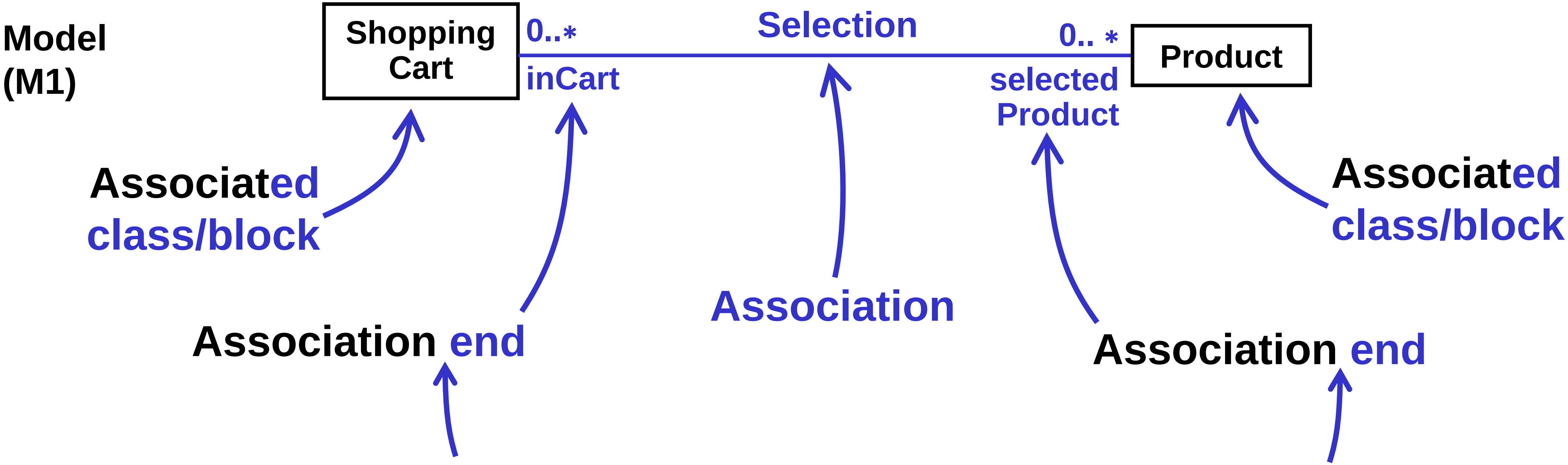
Overview

- Associations
- Debt (SysML1 & SST)
  - Paid (SST February)
  - Unresolved (still paying interest)
- Proposals
- Summary

Overview

- Associations
- Debt (SysML1 & SST)
  - Paid (SST February)
  - Unresolved (still paying interest)
- Proposals
- Summary

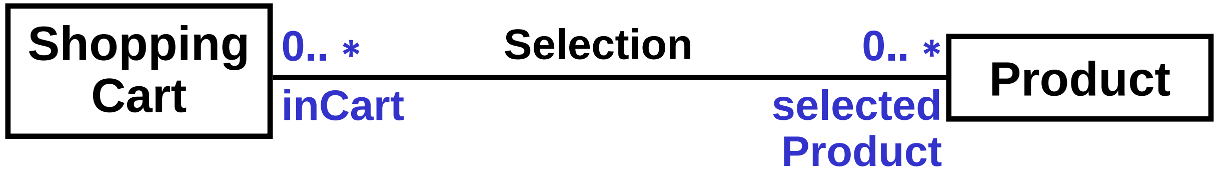
UML/SysML1 Associations



- Association ends are properties for identifying
  - ("navigating to") instances of one associated class/block …
  - … based on ("from") instances of the other.

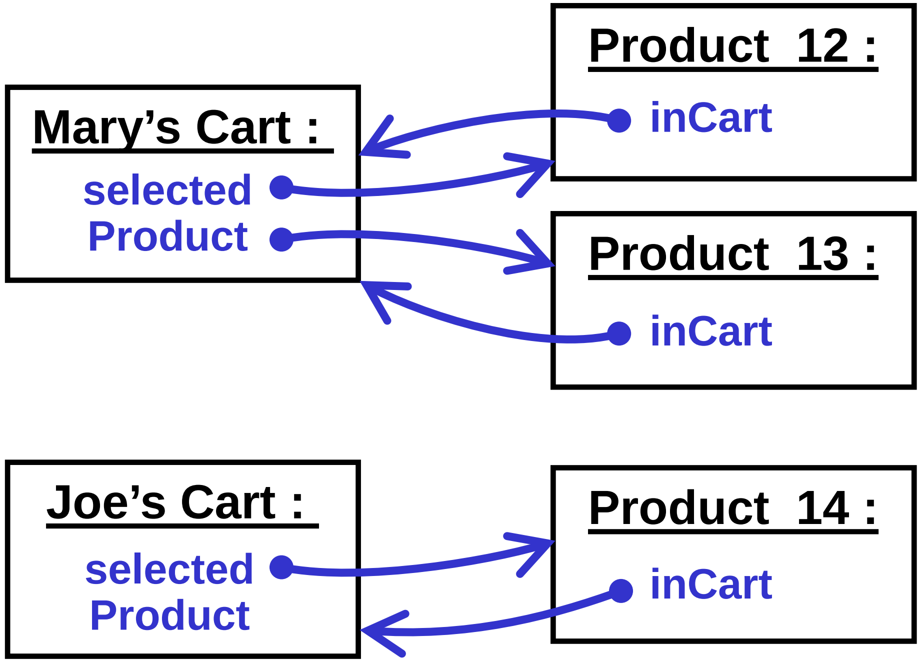
Association “Ends”

Model
(M1)



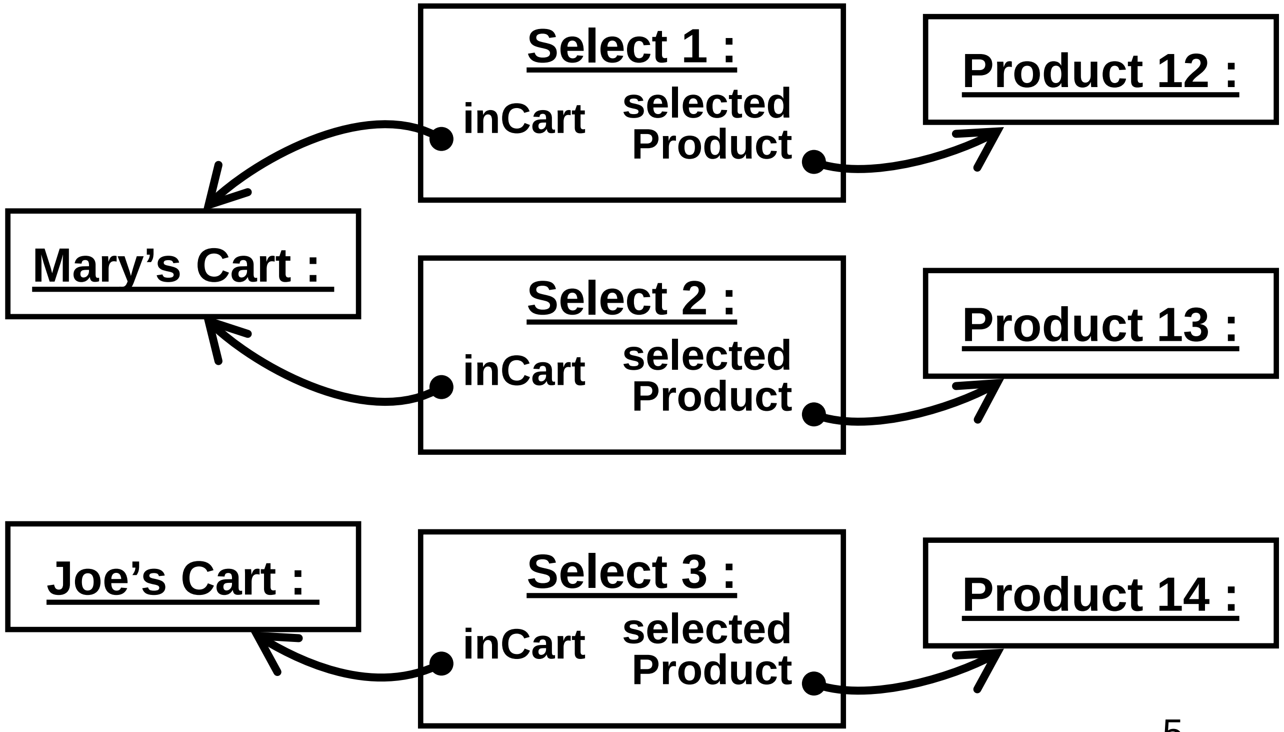
Things
Being
Modeled
(M0)

Object oriented
(UML/SysML1)



Modelled “in” associated classes/blocks

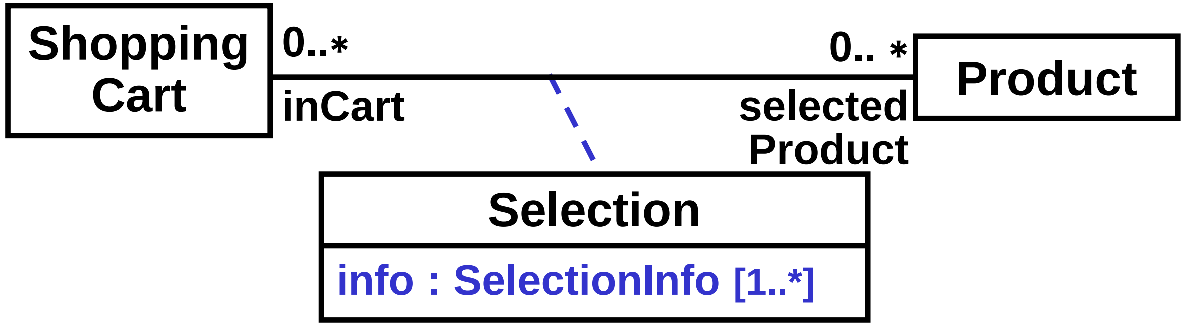
Relational



Modelled from the “middle”

Relational: Properties of Associations

Model
(M1)



Shopping Cart

0..*

inCart

0.. *

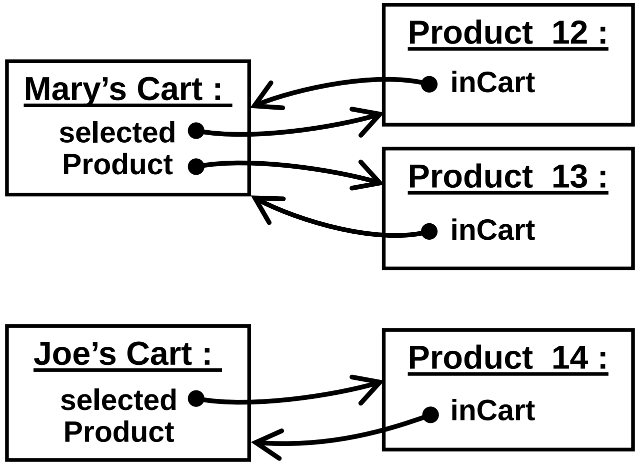
selected Product

Product

Selection

info : SelectionInfo [1..*]

Object oriented



Mary's Cart :

selected Product

Product 12 :

inCart

Product 13 :

inCart

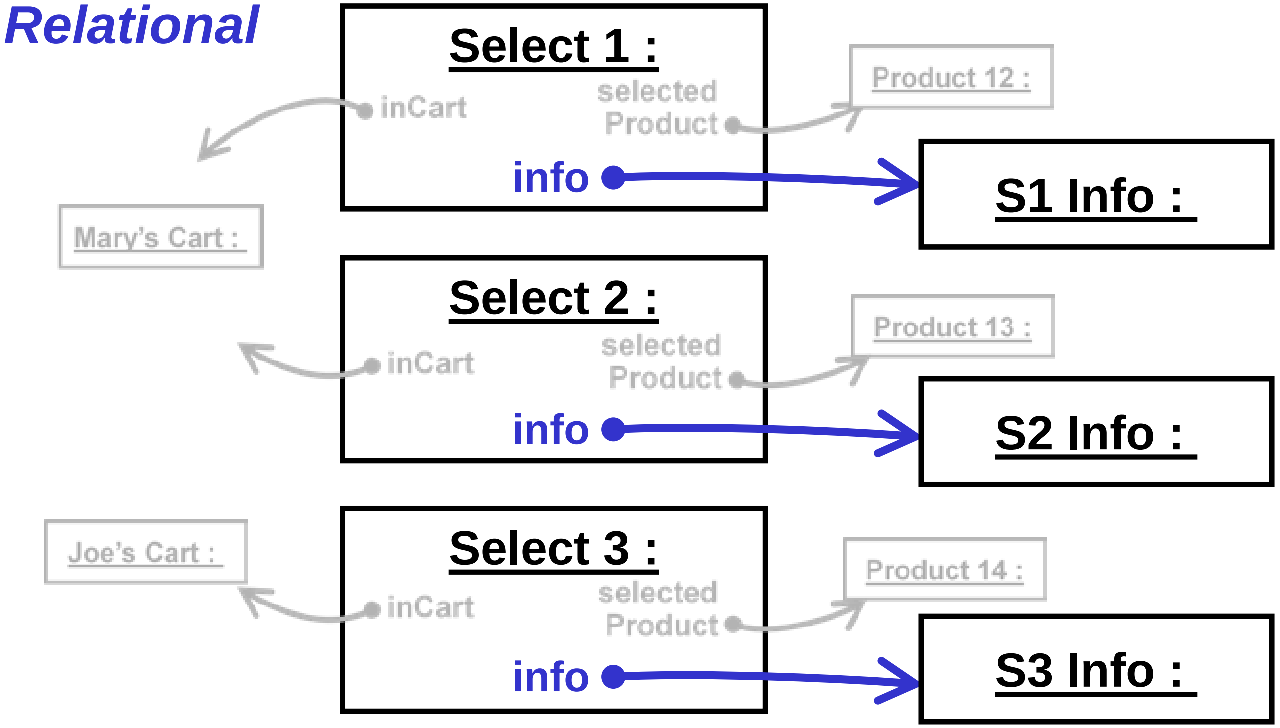
Joe's Cart :

selected Product

Product 14 :

inCart

Relational



Select 1 :

inCart

selected Product

info

Product 12 :

S1 Info :

Mary's Cart :

Select 2 :

inCart

selected Product

info

Product 13 :

S2 Info :

Joe's Cart :

Select 3 :

inCart

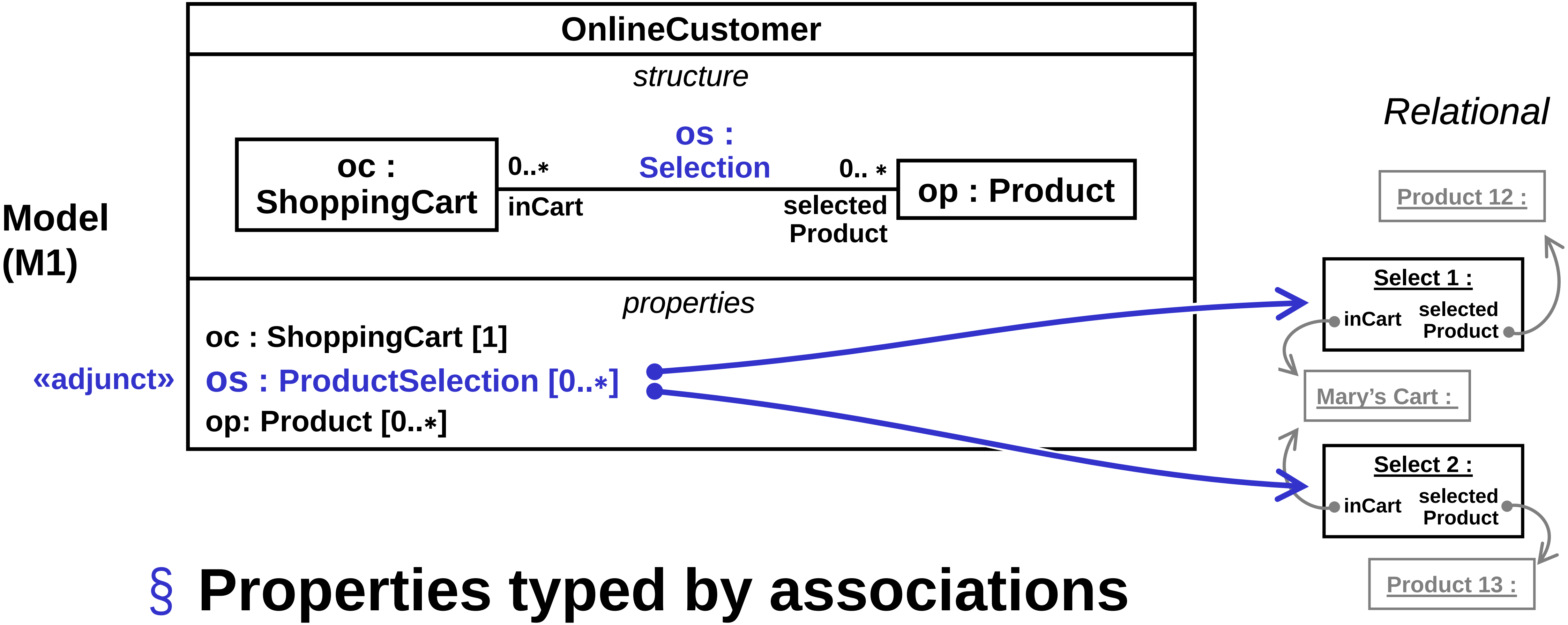
selected Product

info

Product 14 :

S3 Info :

Relational: Connectors



Model (M1)

OnlineCustomer

structure

oc : ShoppingCart

0..* inCart

os : Selection

0.. * selected Product

op : Product

properties

oc : ShoppingCart [1]

«adjunct» os : ProductSelection [0..*]

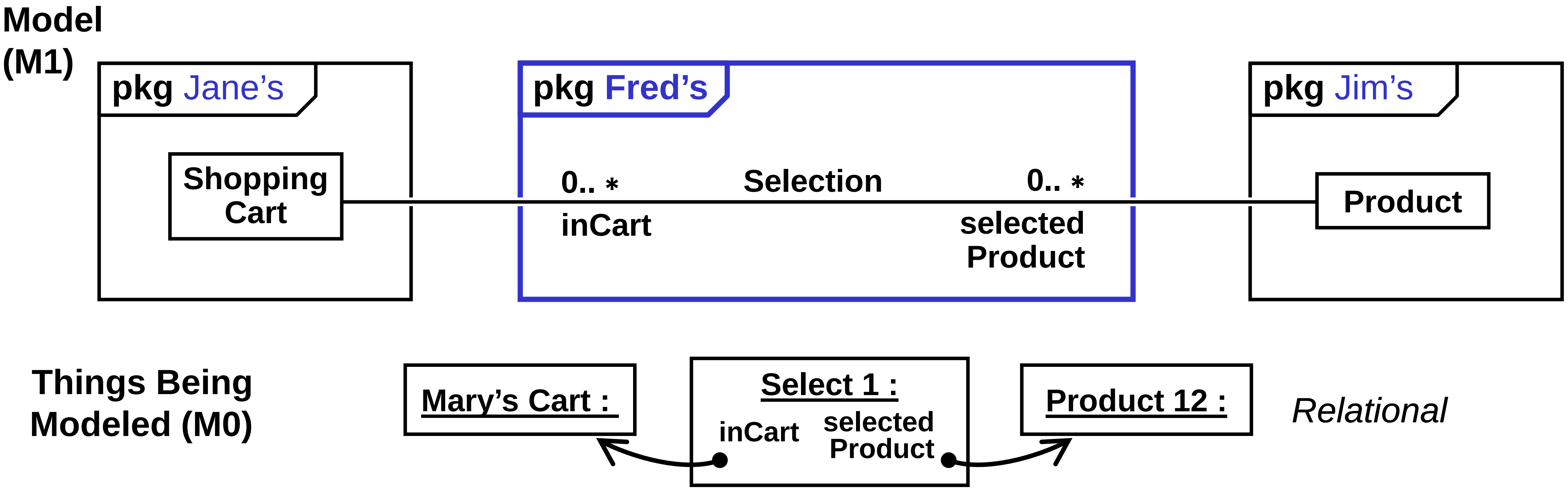
op: Product [0..*]

Relational

§ **Properties typed by associations**

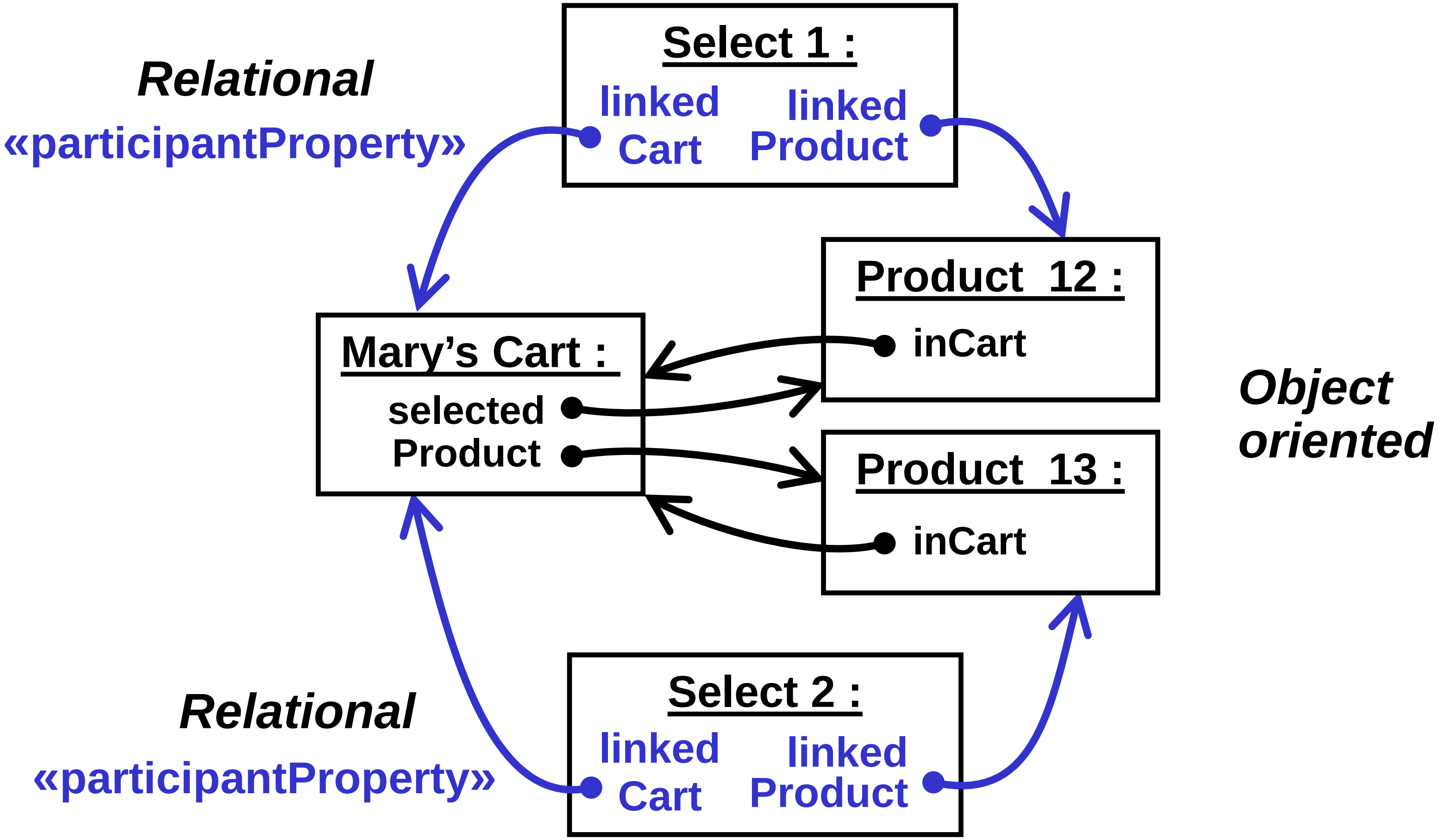
Things Being Modeled (M0)

Relational: “Blind” Association



- Associating classes/blocks that you don’t “own”.
  - Ends are not “in” associated classes/blocks …
  - … but still navigable.
  - UML/SysML1 association-owned ends.

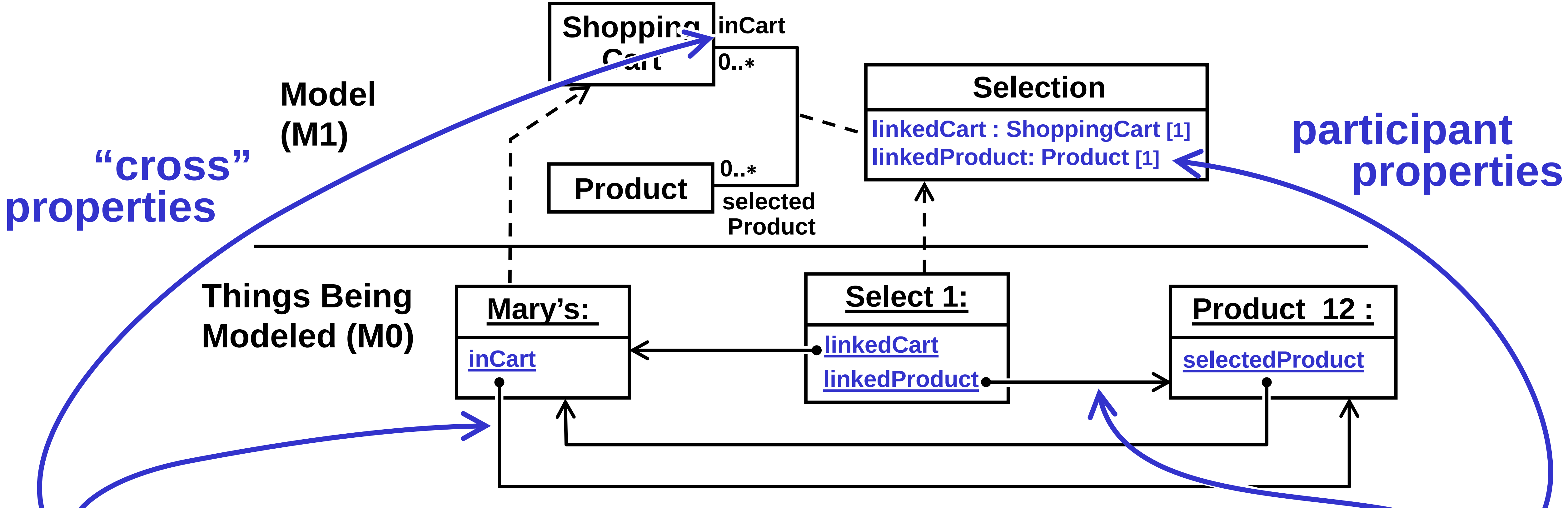
SysML1 Association Participant Properties



Ref roles papers

- § SysML 1 **ParticipantProperty** stereotype.
  - – Always multiplicity [1] {readonly}

Two Kinds of Association Properties

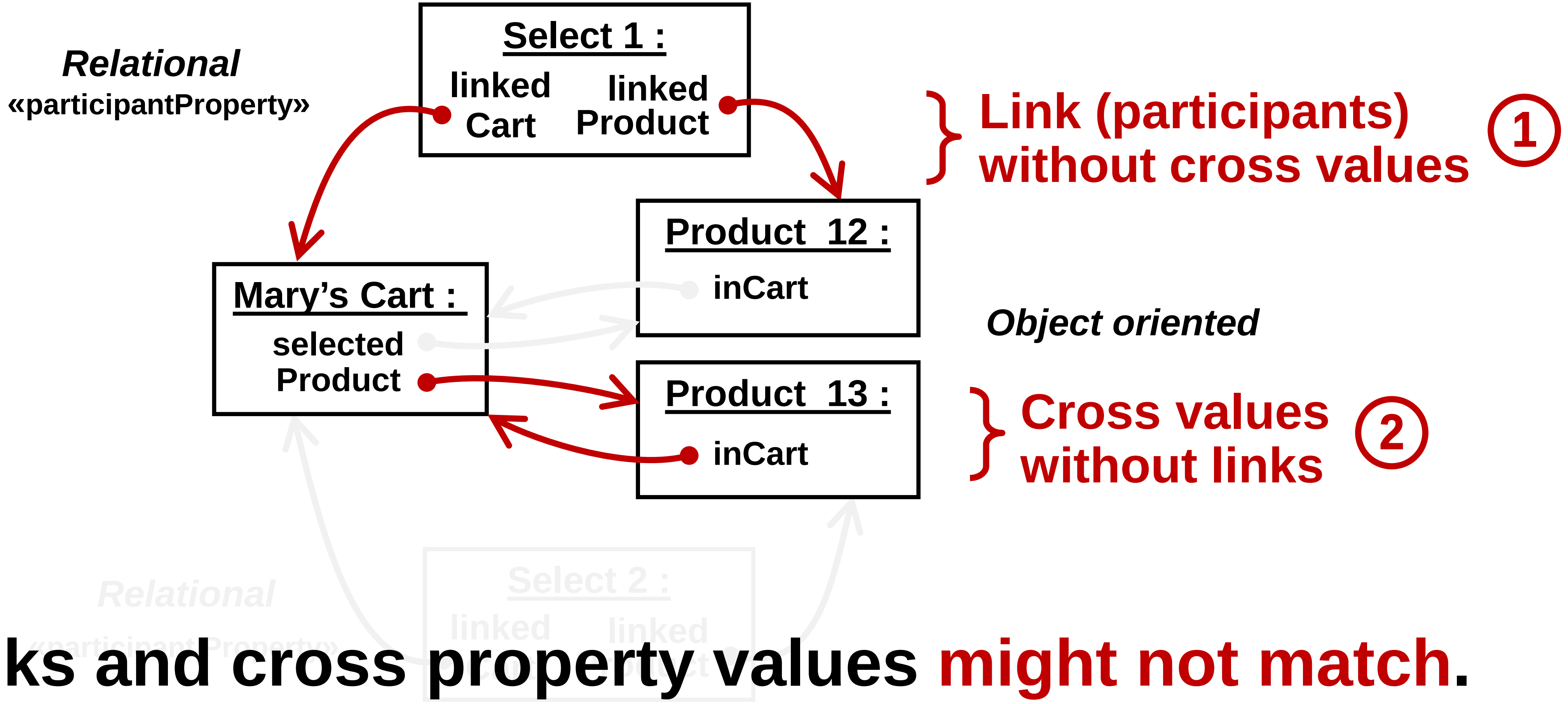


- **Two kinds of properties, for navigating from:**
  - One linked thing to another ("cross" properties).
  - Identify linked things (participant properties).
    - Exactly one thing for each, cannot change.

Overview

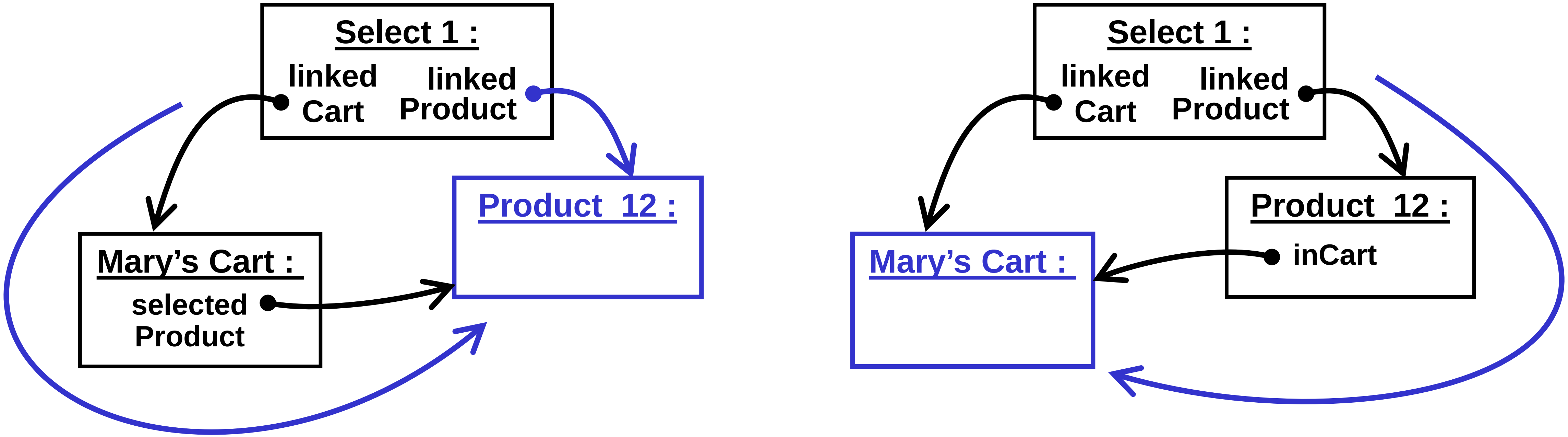
- Associations
- **Debt (SysML1 & SST)**
  - **Paid (SST February)**
  - Unresolved (still paying interest)
- Proposals
- Summary

(In) Consistency



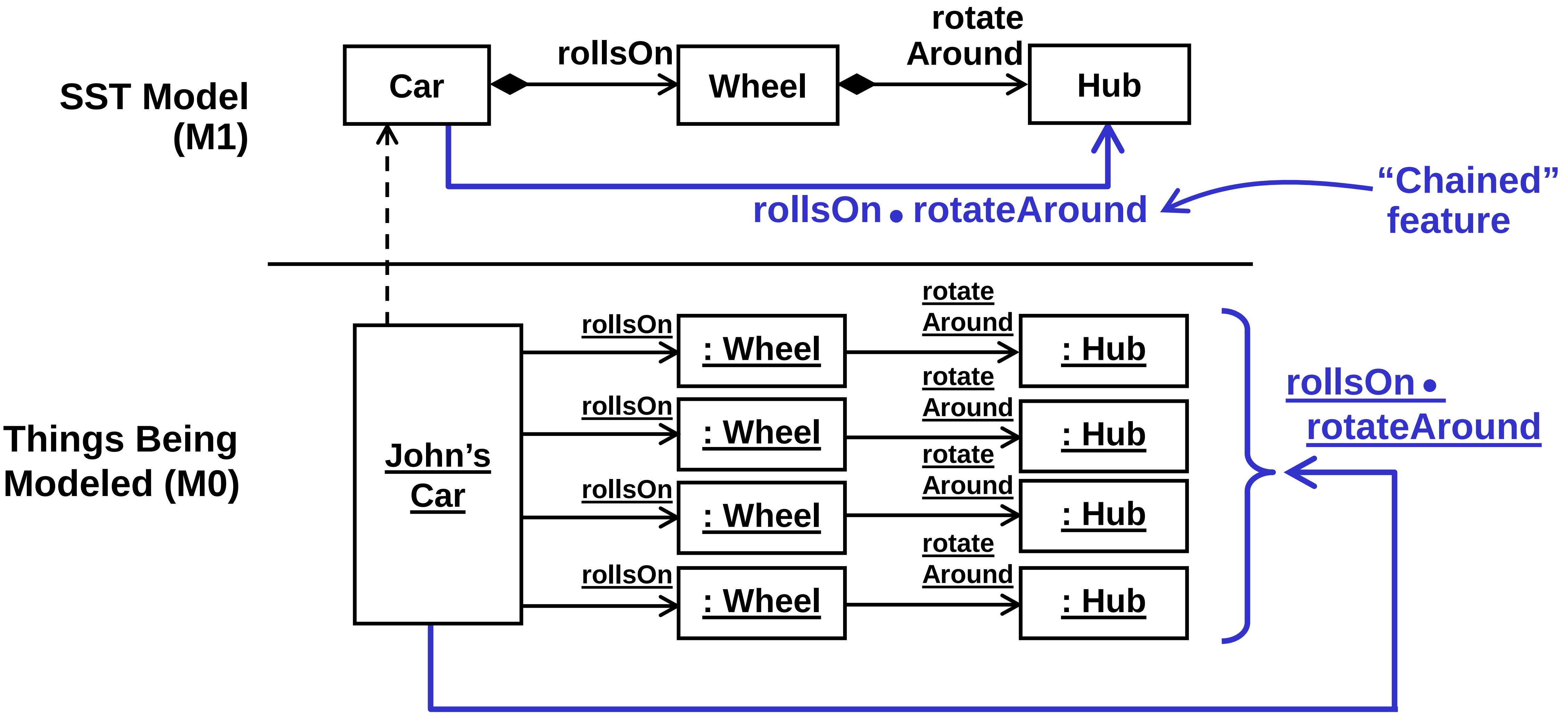
- Links and cross property values might not match.
  - How to require them to?

~~(In)~~ Consistency (1)



- Navigating "out" one participant property …
- … then along a cross property …
- … should be possible to get back to the other participant.

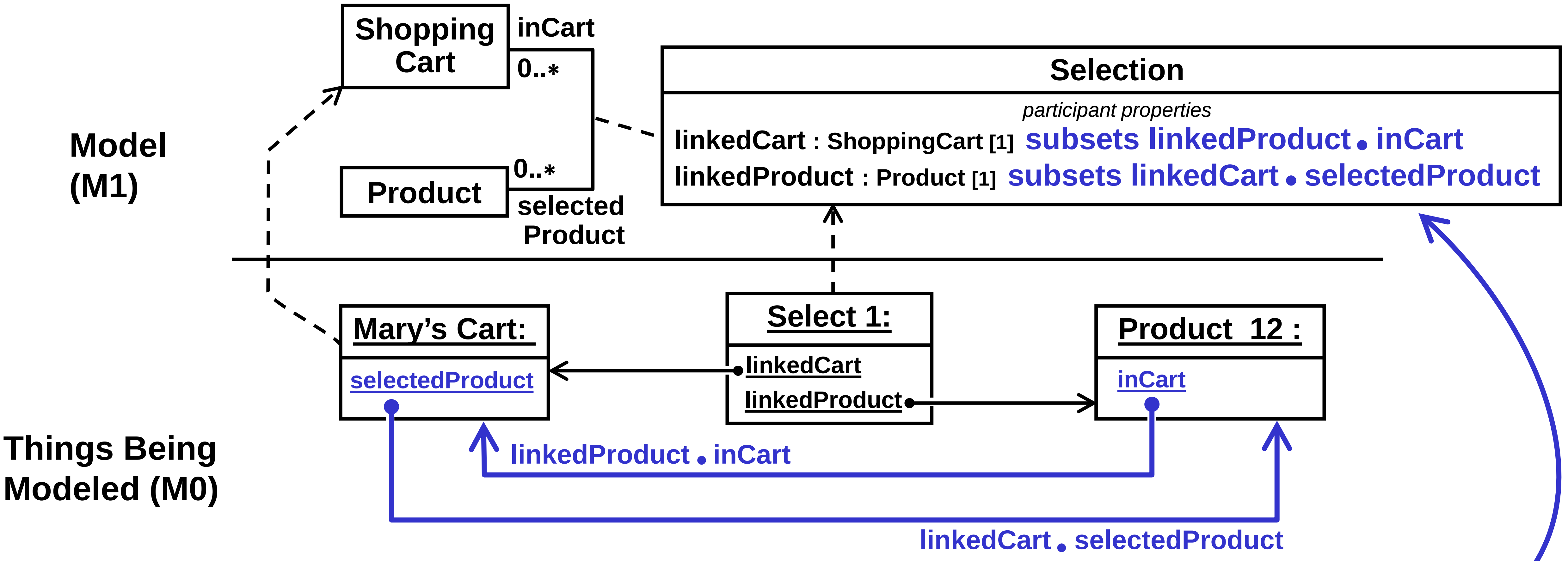
SysML 1.x Bound Refs / SST Feature Chains



§ ≡ Math relational composition

https://en.wikipedia.org/wiki/Composition_of_relations

Consistency (1)



Shopping Cart

inCart

0..*

Product

0..*

selected Product

Selection

participant properties

linkedCart : ShoppingCart [1] subsets linkedProduct • inCart

linkedProduct : Product [1] subsets linkedCart • selectedProduct

Model (M1)

Things Being Modeled (M0)

Mary's Cart:

selectedProduct

Select 1:

linkedCart

linkedProduct

Product 12 :

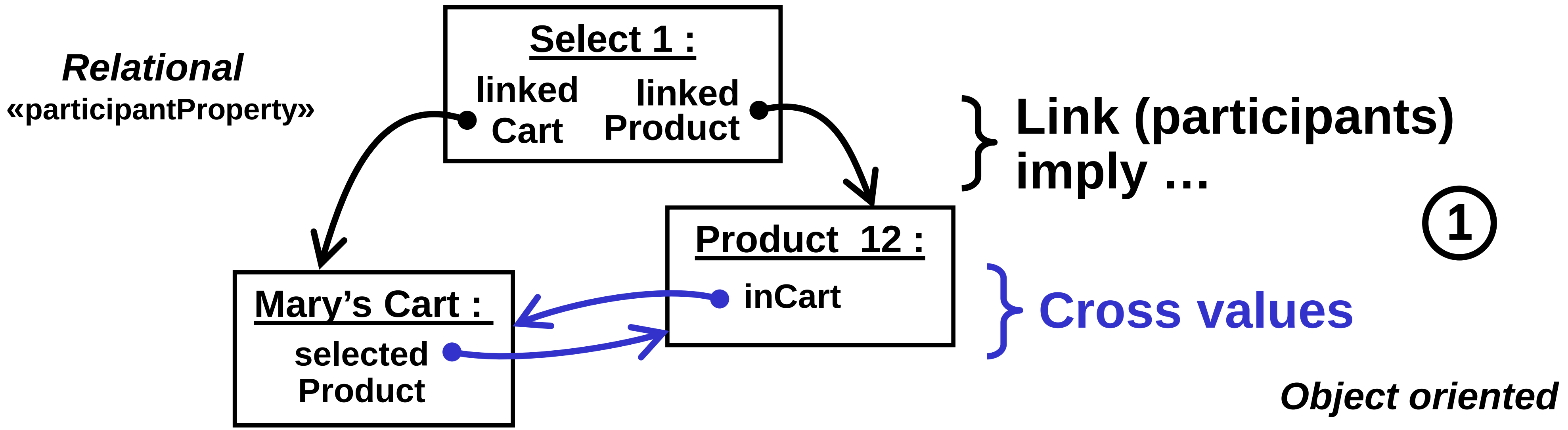
inCart

linkedProduct • inCart

linkedCart • selectedProduct

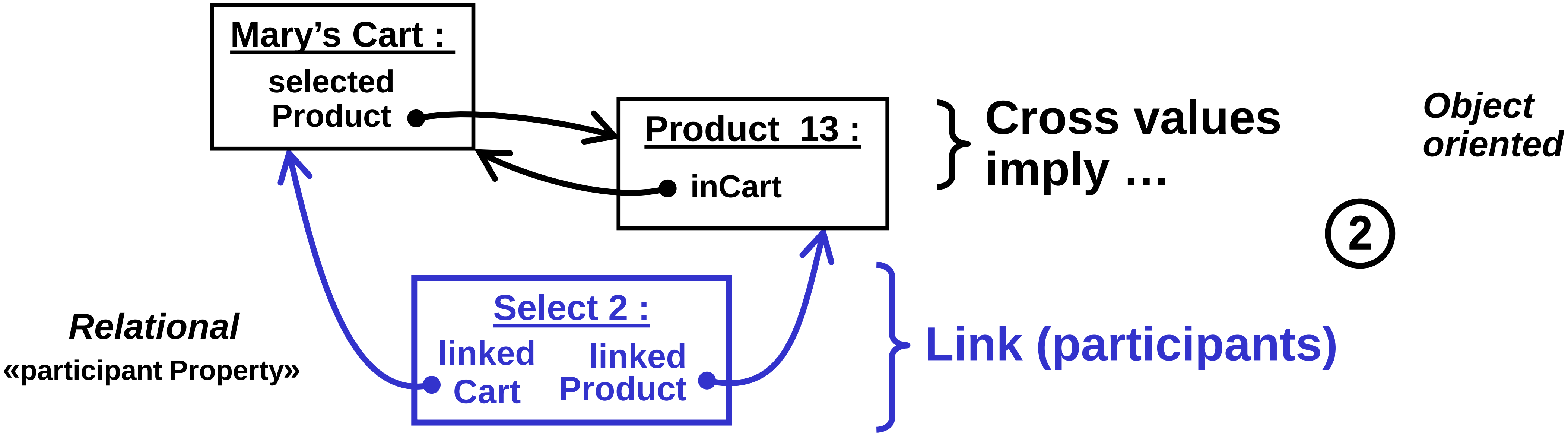
- § **Each participant property subsets across the other.**
  - – **Link (with participants) implies cross property values.**

Consistency (1)



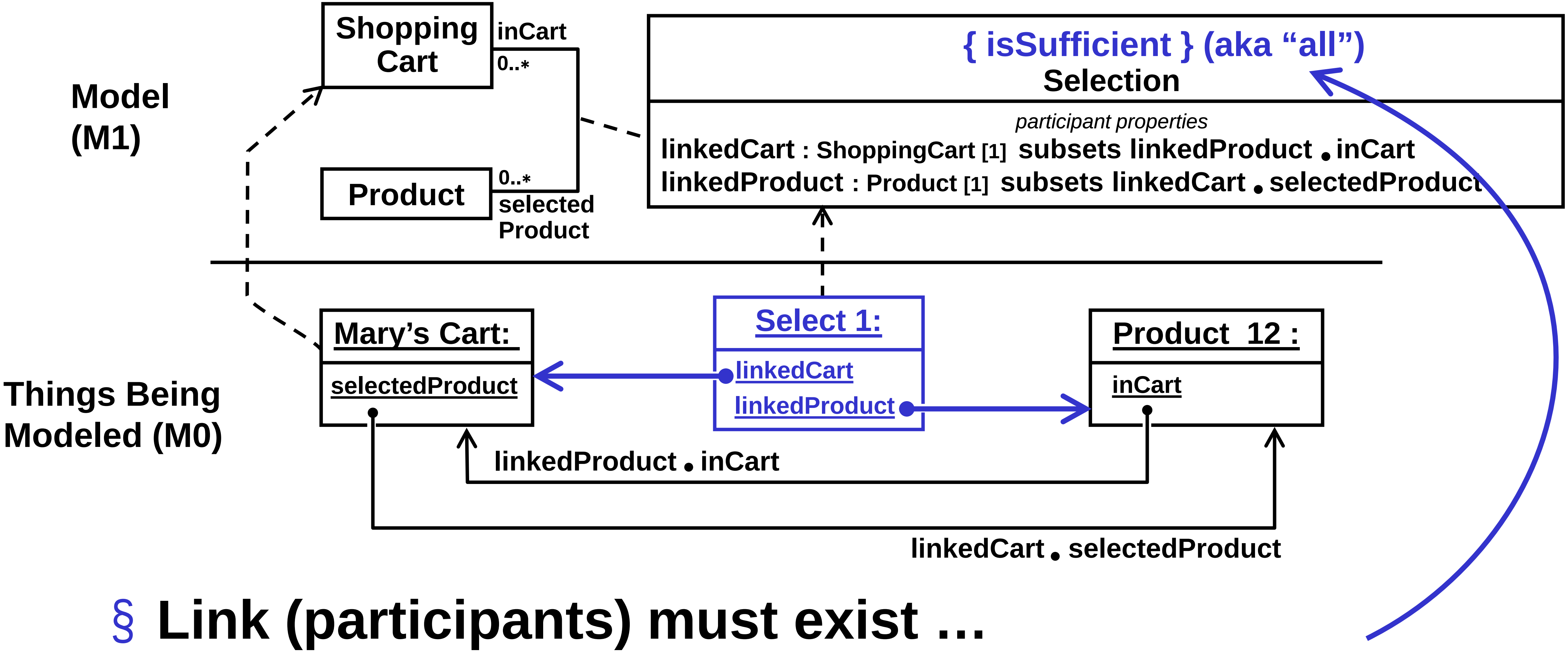
- Cross values required to match link participants.
  - by subsetting participants by cross properties.

(In) Consistency (2)



- Link (participants) required to match cross values.
  - How? Cross values chained through links?

~~(In)~~ Consistency (2)


Model
(M1)
Shopping Cart
inCart
0..*
Product
0..*
selected Product
{ isSufficient } (aka “all”)
Selection
participant properties
linkedCart : ShoppingCart [1] subsets linkedProduct . inCart
linkedProduct : Product [1] subsets linkedCart . selectedProduct
Things Being Modeled (M0)
Mary’s Cart:
selectedProduct
Select 1:
linkedCart
linkedProduct
Product 12 :
inCart
linkedProduct . inCart
linkedCart . selectedProduct

§ **Link (participants) must exist …**

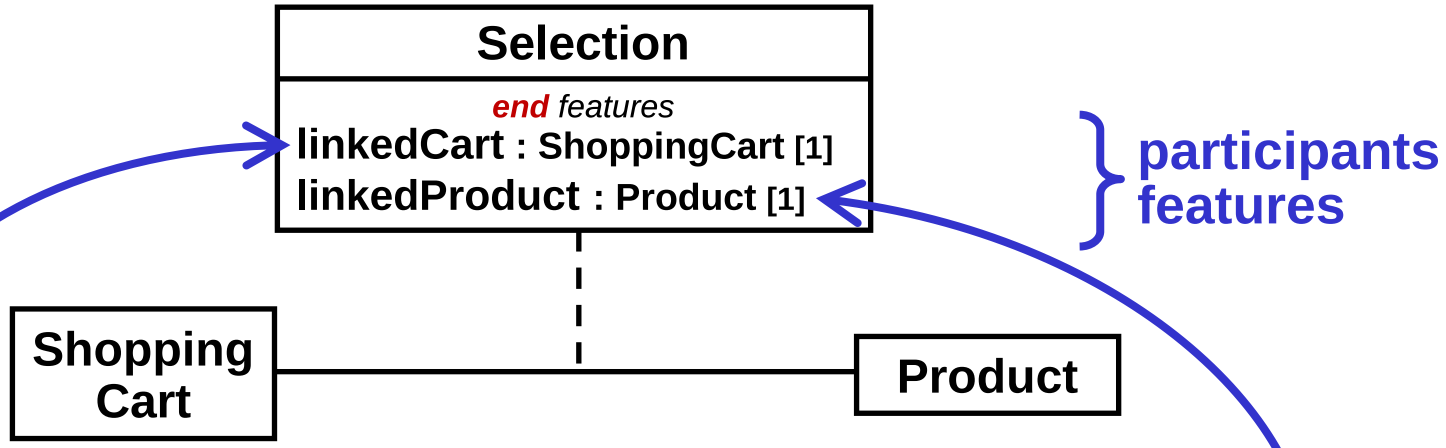
– **… when cross values do (subsettings are “satisfied”)**

Overview

- Associations
- Debt (SysML1 & SST)
  - Paid (SST February)
  - Unresolved (still paying interest)
- Proposals
- Summary

SST Assoc “Ends” (textual syntax)

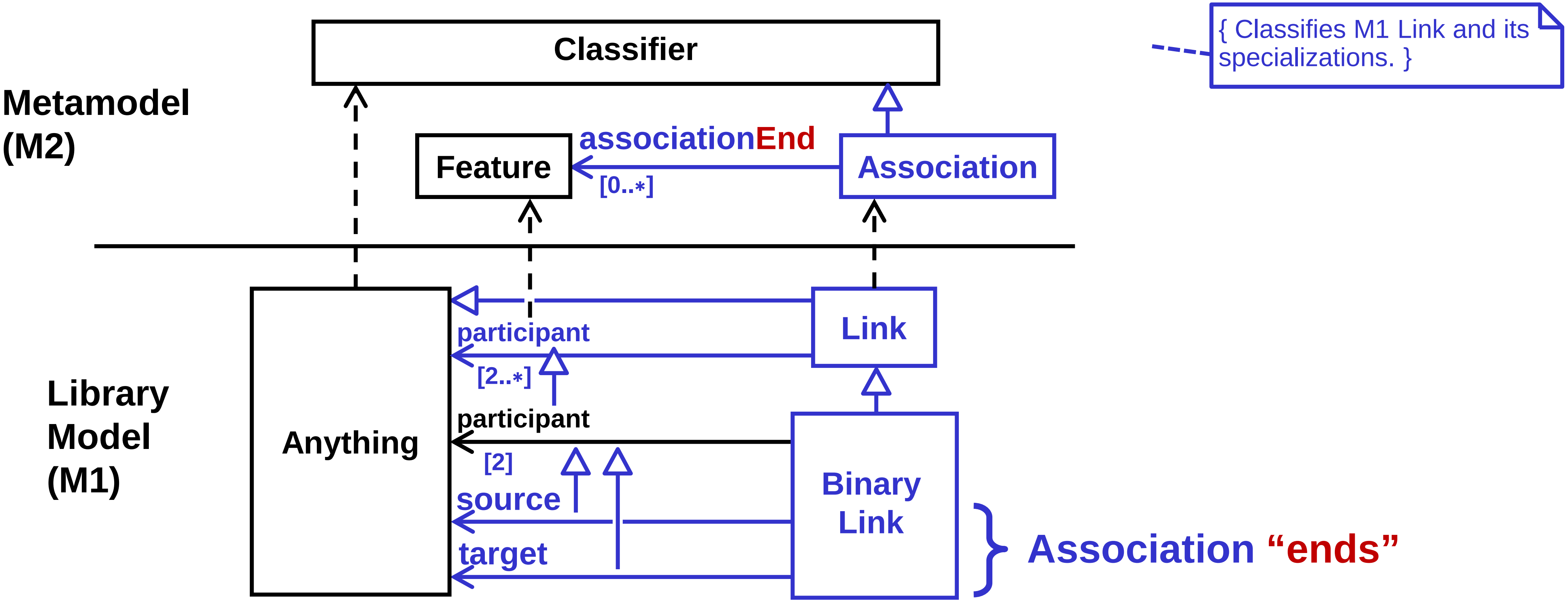
Model
(M1)



```
assoc Selection {
  end linkedCart: ShoppingCart;
  end linkedProduct: Product; }
```

- SST “end” features are actually participant features.
  - Values are the things being linked, exactly one each.
  - Better term?   Needs to be short, not an abbreviation.

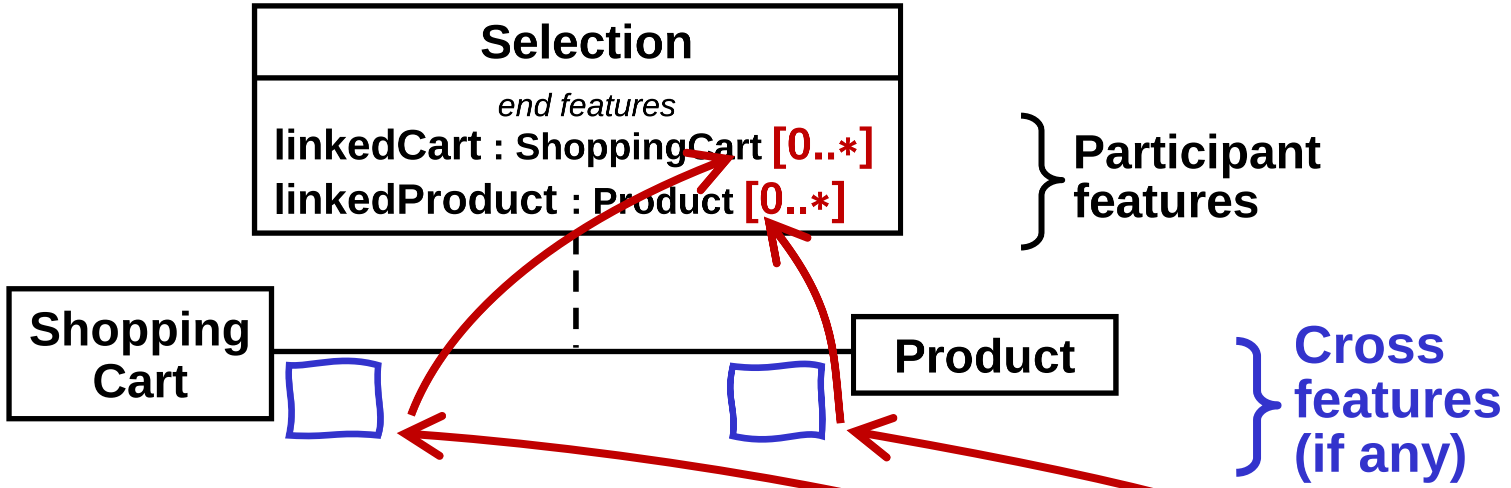
SST Assoc “Ends” (abstract syntax)



- Same term in the metamodel.

SST Assoc Textual Syntax (multiplicity)

Model
(M1)



Selection

end features

linkedCart : ShoppingCart [0..*]

linkedProduct : Product [0..*]

Participant features

Shopping Cart

Product

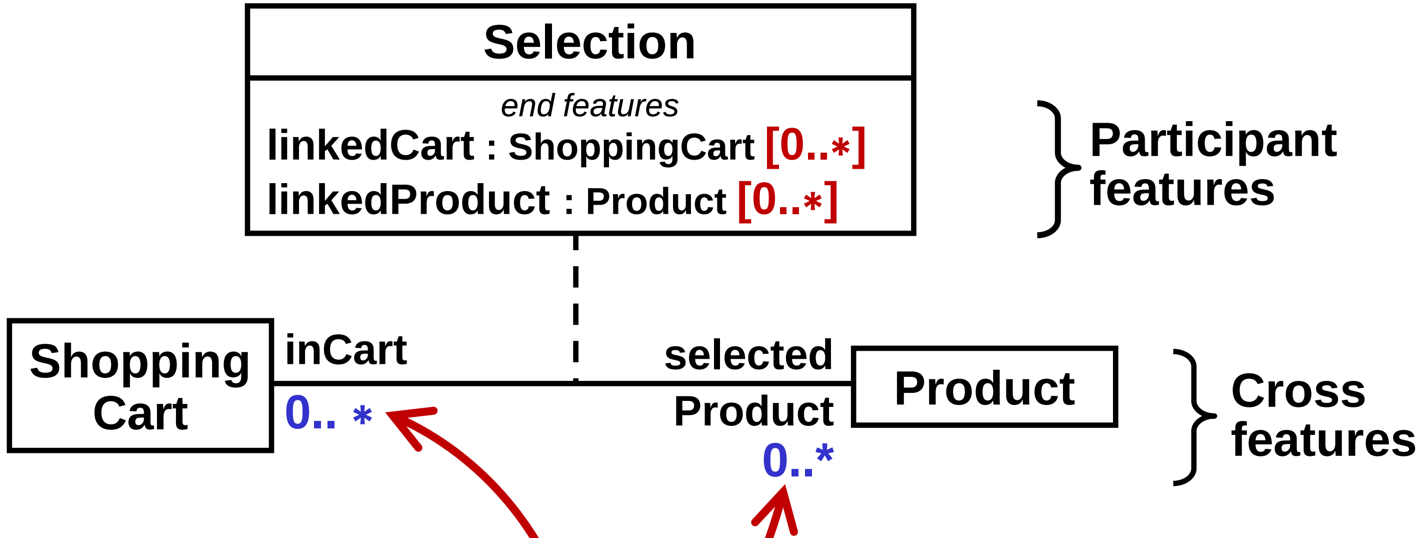
Cross features (if any)

```
assoc Selection {
   end linkedCart: ShoppingCart[0..*];
   end linkedProduct: Product[0..*];}
```

- § “End” multiplicity is about cross features …,
- § … which might not exist.
  - – Useful for one-way and non-navigable associations.
- § Looks like participant multiplicity (if you know “end”).

SST Assoc Textual Syntax (multiplicity)

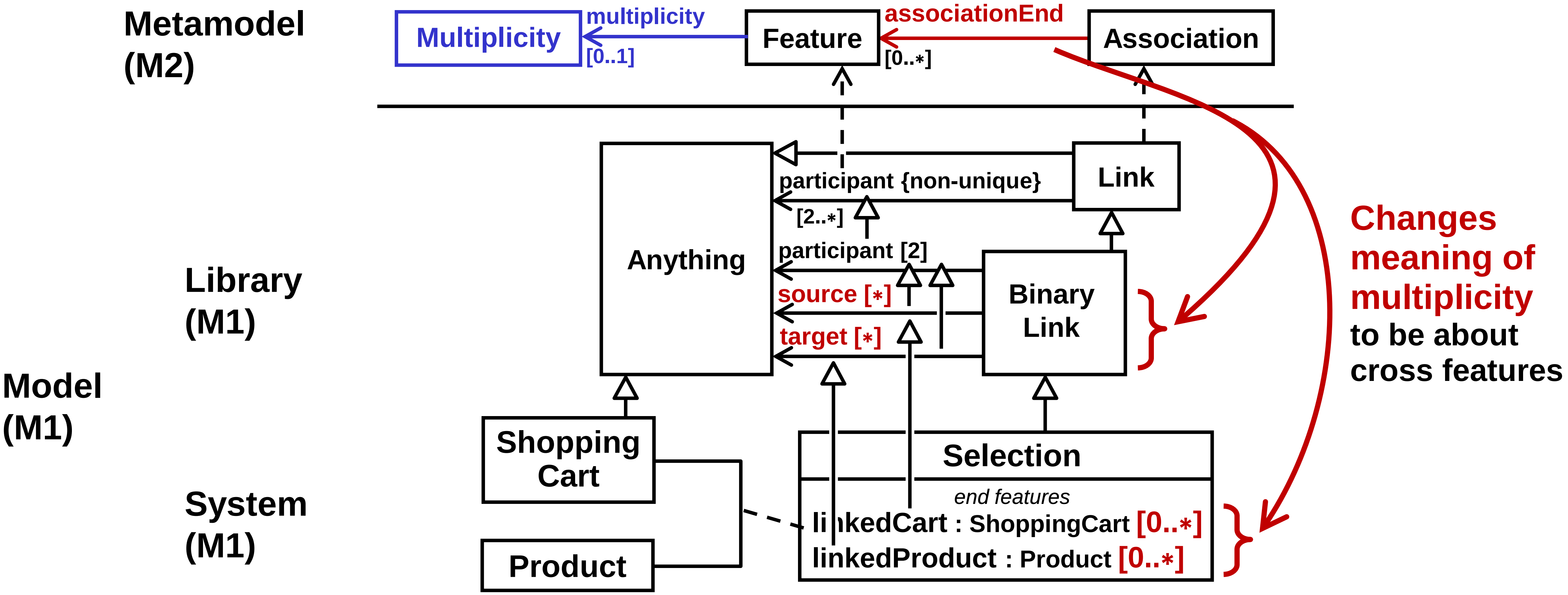
Model
(M1)



```
assoc Selection {
  end linkedCart: ShoppingCart[0..*] subsets linkedProduct.inCart;
  end linkedProduct: Product[0..*] subsets linkedCart.selectedProduct;}
```

- “End” multiplicity redundant and possibly conflicting
  - between “end” and cross features.
- In textual and abstract syntax.

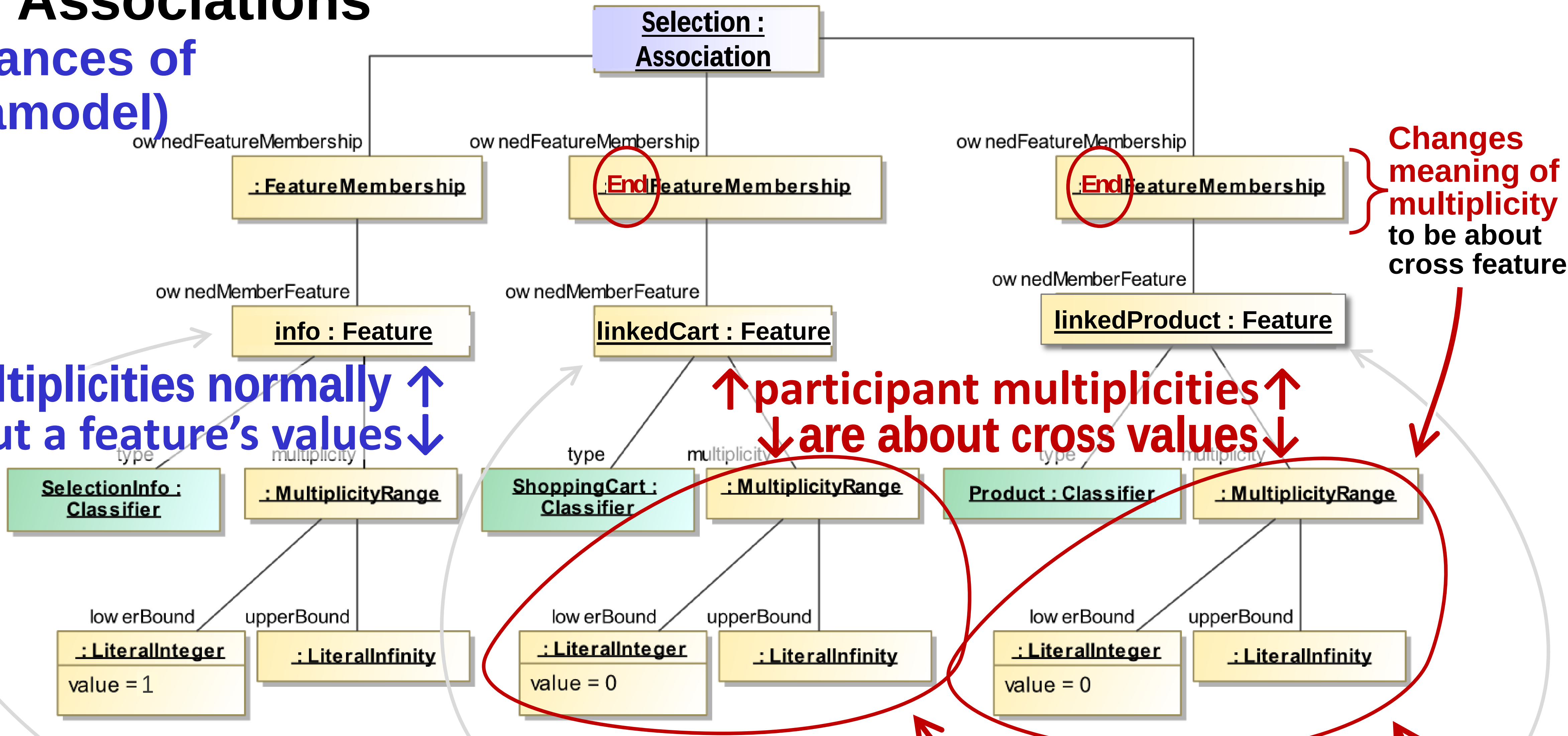
SST Assoc Abstract Syntax (multiplicity)



- **"End" changes feature multiplicity semantics …**
  - … to be about cross features.
  - Number of participant feature values not restricted.

SST Associations

(instances of metamodel)



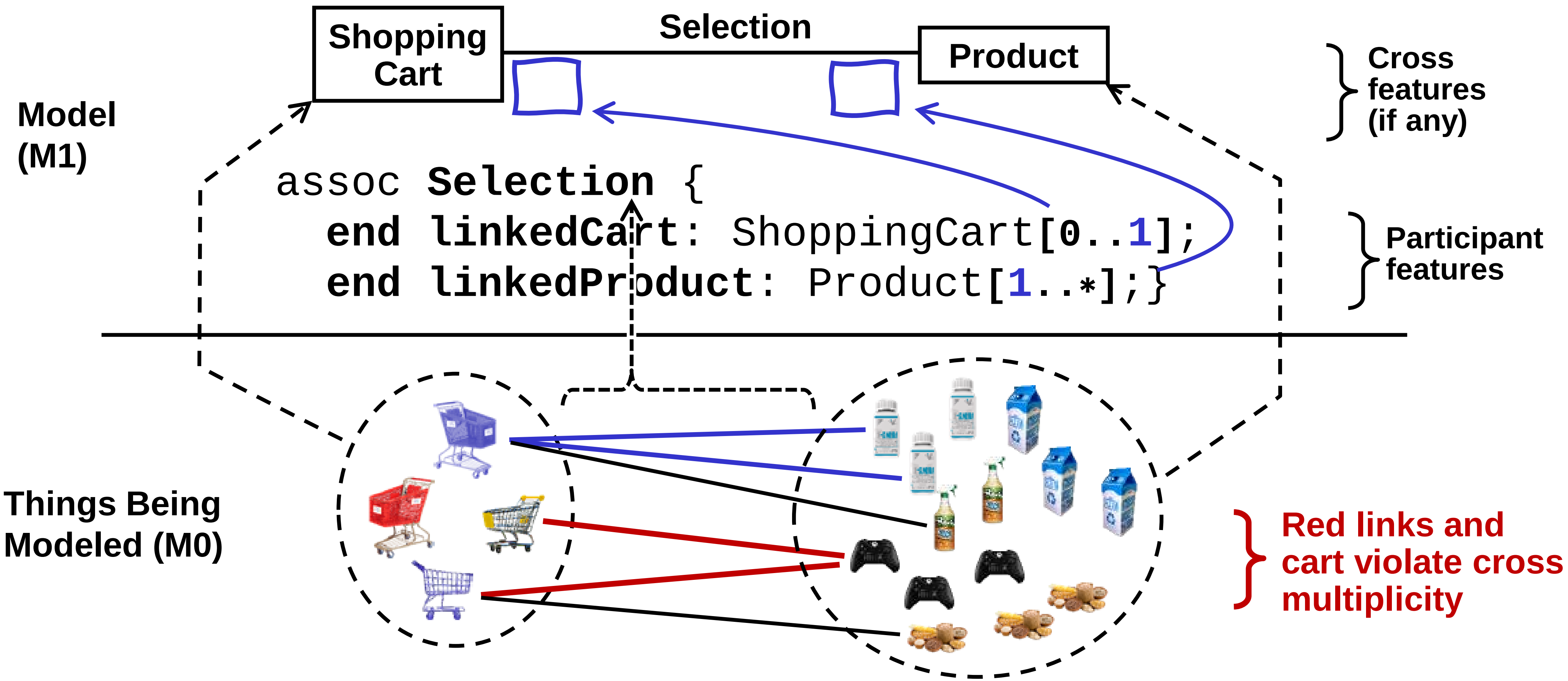
Changes meaning of multiplicity to be about cross features

multiplicities normally ↑ about a feature's values ↓

↑participant multiplicities↑ ↓are about cross values↓

```
assoc Selection {
  end cart: ShoppingCart[0..*];
  end selectedProduct: Product[0..*];
  feature info: SelectionInfo [1..*]; }
```

Assoc Semantics (cross multiplicity)



Selection

Shopping Cart

Product

Model (M1)

Cross features (if any)

```
assoc Selection {
  end linkedCart: ShoppingCart[0..1];
  end linkedProduct: Product[1..*];}
```

Participant features

Things Being Modeled (M0)

Red links and cart violate cross multiplicity

- Every cart and product must satisfy cross multiplicity.
  - via Selection links.

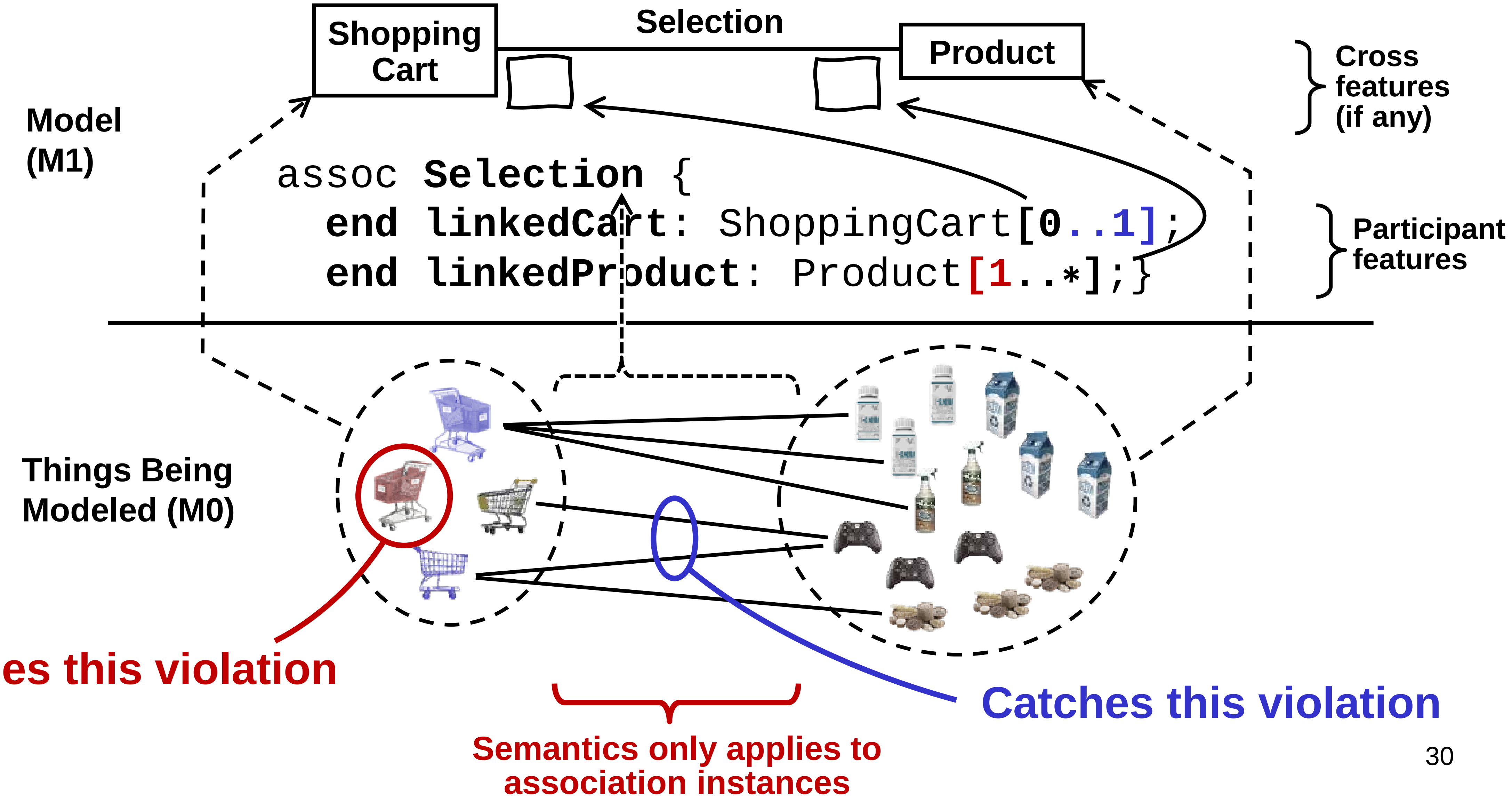
SST Assoc Semantics (cross multiplicity)

- **Informal text currently.**
- **Applies only to instances of association (links),**
  - **rather than instances of the associated classifiers.**

[7.4.5] if an association end has a multiplicity specified other than 1..1, then this is interpreted as follows: *For each association end, the multiplicity, ordering and uniqueness constraints specified for that end apply to each set of instance of the association that have the same (single) values for each of the other ends.* For a binary association, this corresponds to the multiplicity resulting from "navigating" across the association given a value at one end of the association to the other end of the association.

[8.4.4.5.1] If an associationEnd has a declared multiplicity other than 1..1, then this shall be interpreted as follows: For an Association with *N* associationEnds, consider the *i*-th associationEnd e_i. *The multiplicity, ordering and uniqueness constraints specified for* e_i *apply to each set of instances of the Association that have the same (singleton) values for each of the N-1 associationEnds other than* e_i.

SST Assoc Semantics (cross multiplicity)



Model (M1)

Shopping Cart

Selection

Product

Cross features (if any)

```
assoc Selection {
  end linkedCart: ShoppingCart[0..1];
  end linkedProduct: Product[1..*];}
```

Participant features

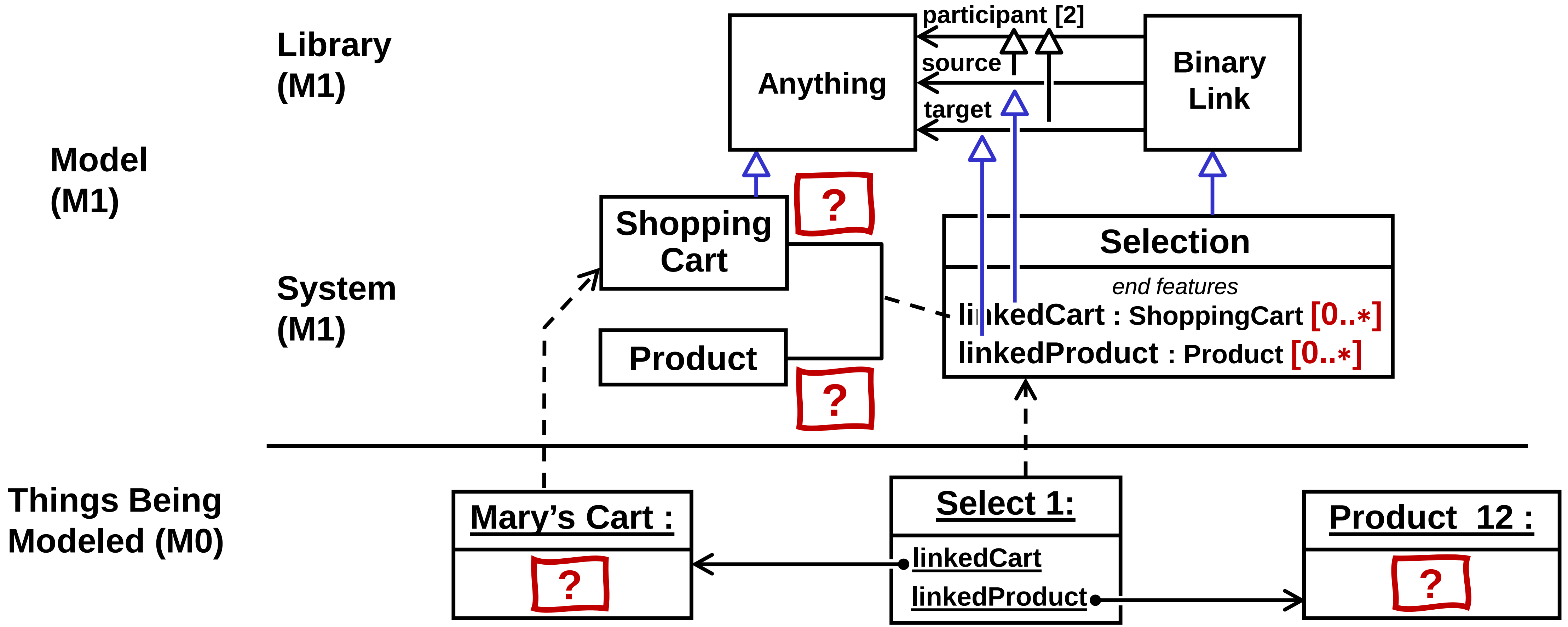
Things Being Modeled (M0)

Misses this violation

Catches this violation

Semantics only applies to association instances

SST Assoc, Identifying Cross Features



- Nothing in the textual or abstract syntax for them.
  - Cross-subsetting chain pattern is modeler option.
  - API access can't depend on it, eg, for automated analysis.

Association Debt Summary

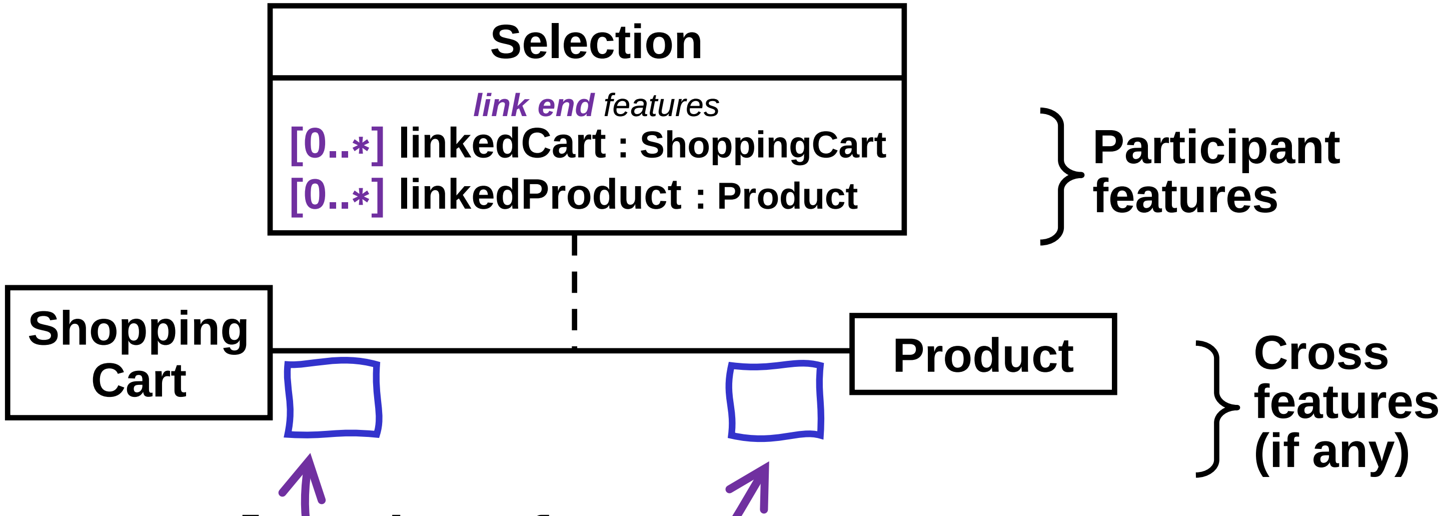
1. Misleading textual syntax
   - Term: “end” for participant rather than cross features
   - Cross feature multiplicity looks like participant’s.
2. Two meanings for feature multiplicity
   - Requires special casing in modeling/analysis tools.
3. Incomplete semantics
   - Number of participant values not restricted (should be [1]).
   - Cross multiplicity semantics is informal and incomplete.
4. Cross features not identified
   - Subsetting pattern is non-standard (tool builders and modelers cannot depend on it).

Overview

- Associations
- Debt (SysML1 & SST)
  - Paid (SST February)
  - Unresolved (still paying interest)
- **Proposals**
- Summary

Assoc Textual Syntax (Proposal)

Model
(M1)



System
(M1)

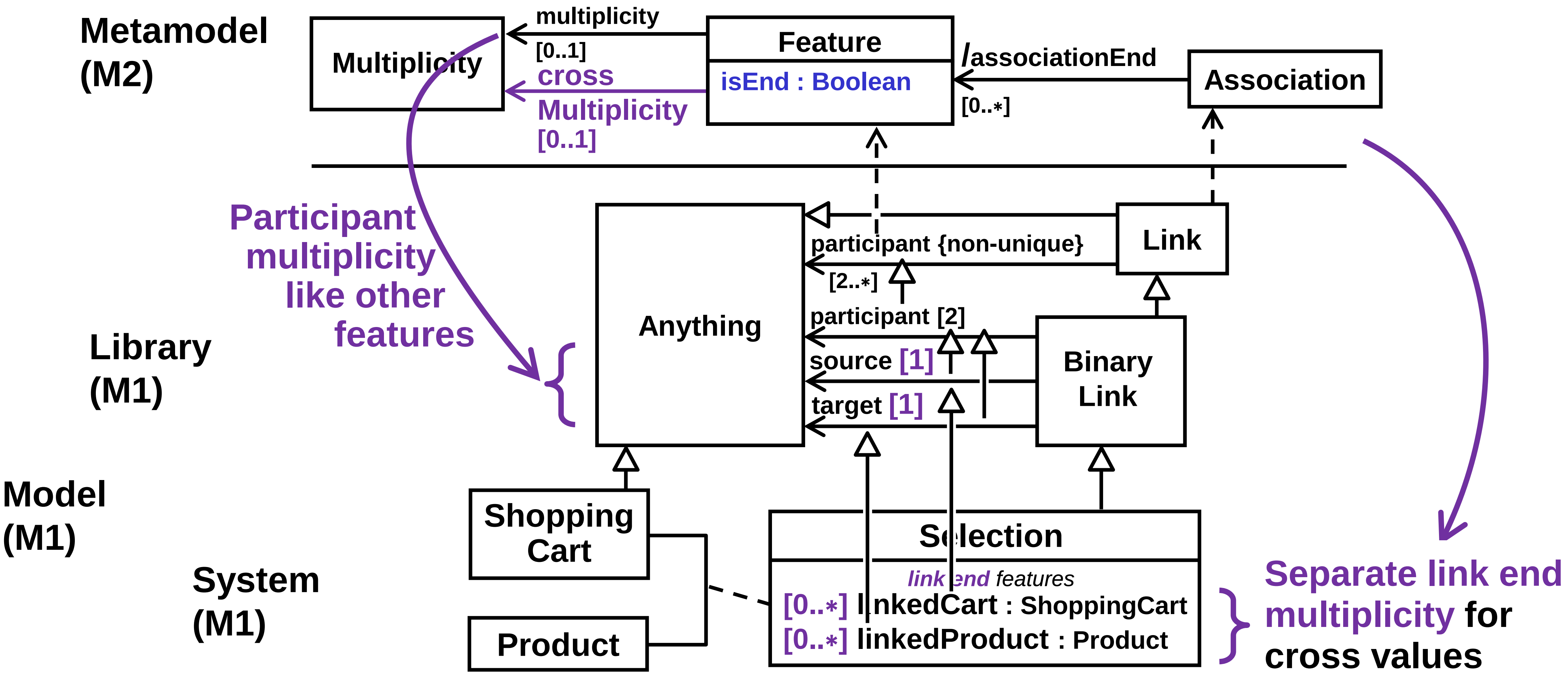
```
assoc Selection {
  linkEnd [0..*] linkedCart: ShoppingCart;
  linkEnd [0..*] linkedProduct: Product;}
```

```
assoc BinaryLink {
  linkEnd source: Anything [1] subsets participant;
  linkEnd target: Anything [1] subsets participant;}
```

Library
(M1)

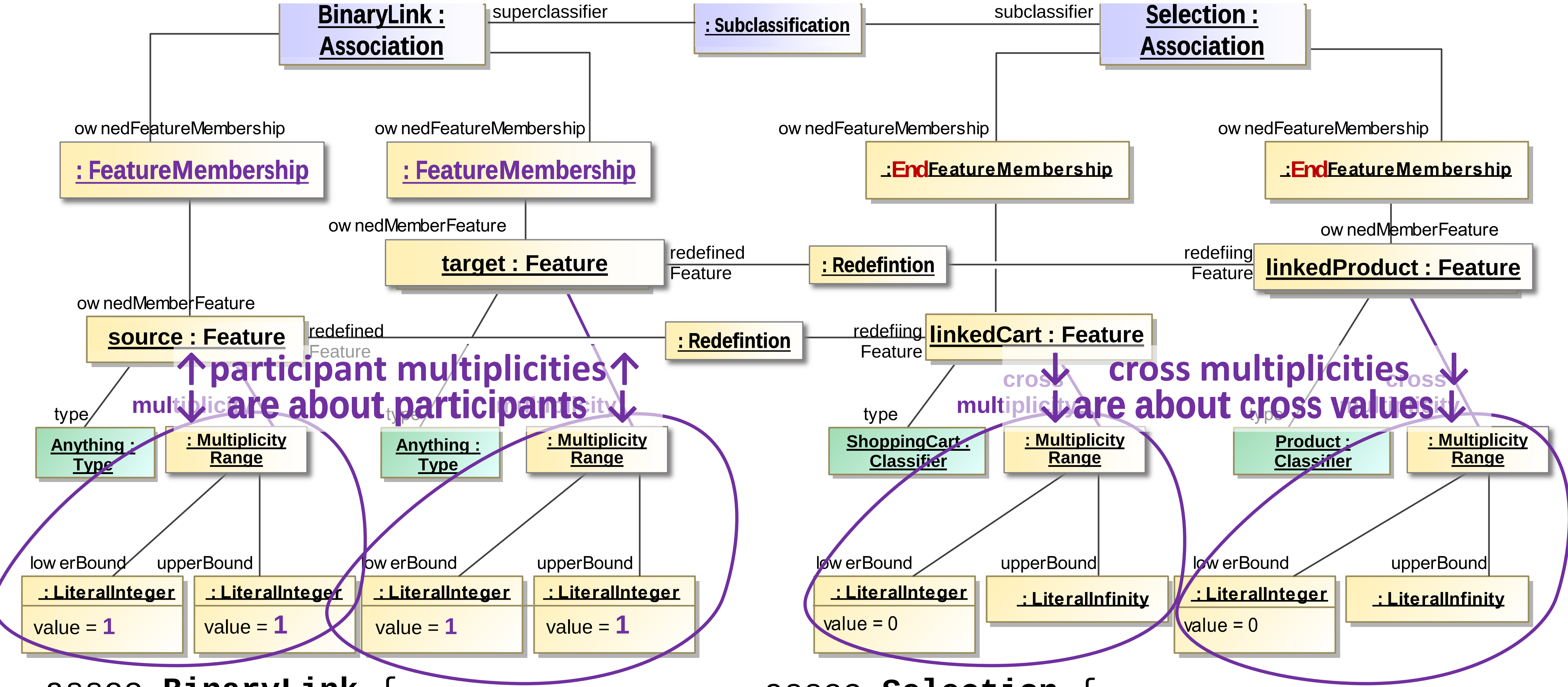
- Keyword and cross multiplicity position change.
  - Leaves (textual) room for participant multiplicity in library.

Assoc Abstract Syntax (Proposal)



§ Cross multiplicity applies only when isEnd=true

Cross multiplicities (instances of metamodel)



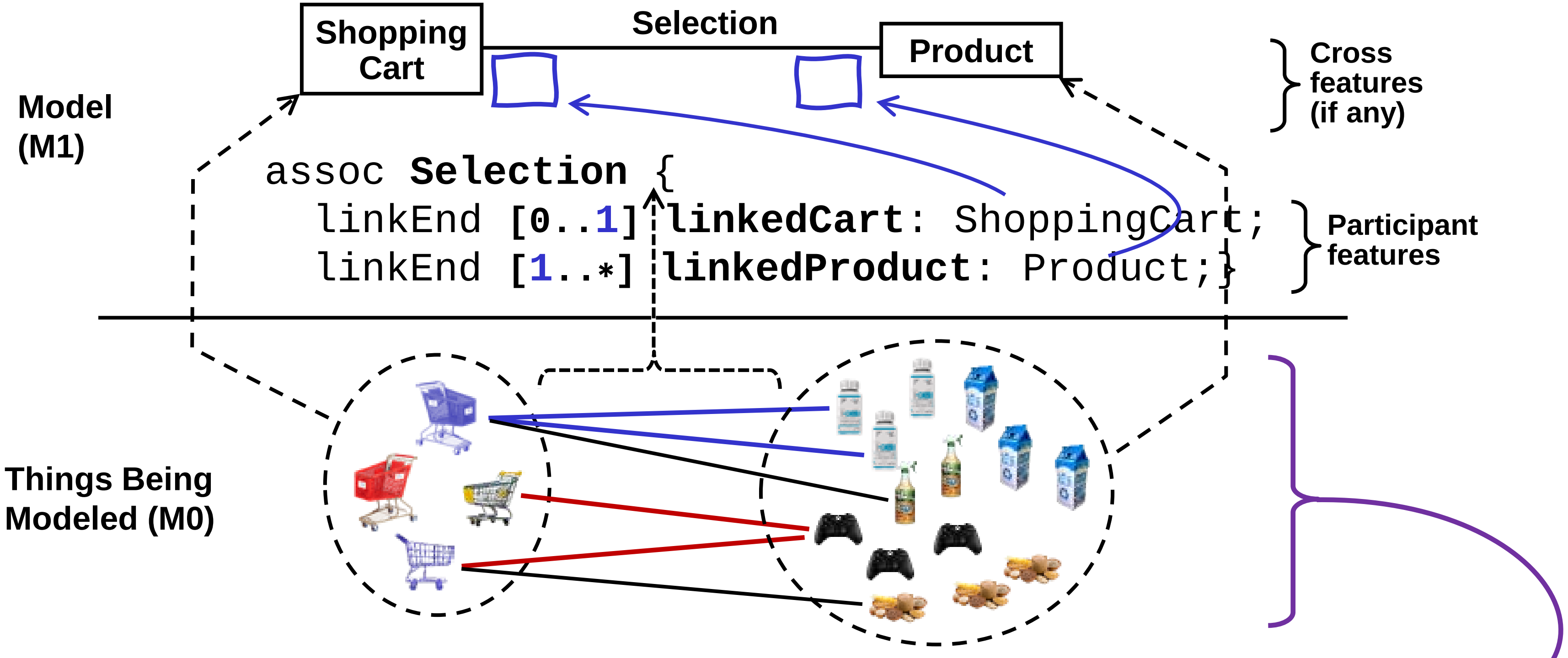
↑participant multiplicities↑
are about participants

cross multiplicities
are about cross values

```
assoc BinaryLink {
   linkEnd source: Anything [1] …;
   linkEnd target: Anything [1] …;}
```

```
assoc Selection {
   linkEnd [0..*] linkedCart: ShoppingCart;
   linkEnd [0..*] linkedProduct: Product;}
```

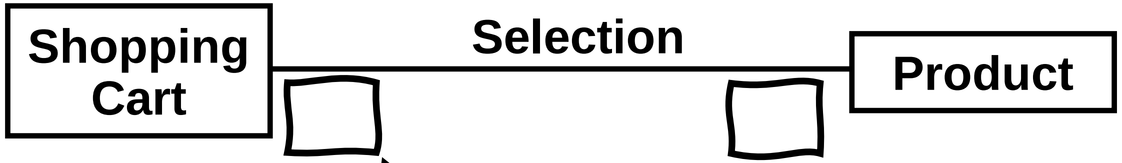
Assoc Semantics (cross mult) (How?)



```
assoc Selection {
  linkEnd [0..1] linkedCart: ShoppingCart;
  linkEnd [1..*] linkedProduct: Product;}
```

- How constrain all instances collectively?
  - Of association and its associated classifiers
- Usually done in math semantics
  - Trying to use math only in Core.

Assoc Semantics (cross mult) (Proposal)



Cross features if any

Model (M1)

```
assoc Selection {
   linkEnd [0..1] linkedCart: ShoppingCart;
   linkEnd [1..*] linkedProduct: Product;}
```

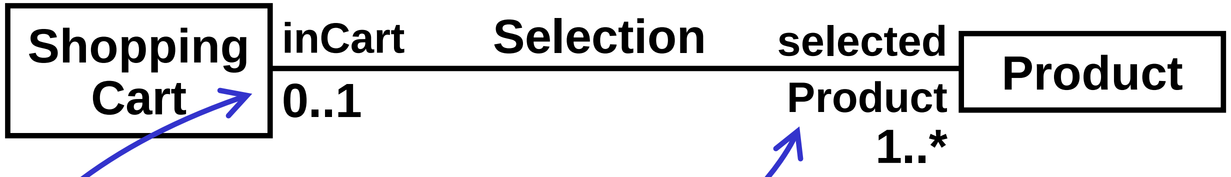
Parses to

```
assoc Selection {
⊕-feature inCart: ShoppingCart [0..1] featured by Product;
⊕-feature selectedProduct: Product [1..*] featured by ShoppingCart;

  linkEnd linkedCart: ShoppingCart
                subsets linkedProduct.(Selection::inCart);
  linkEnd linkedProduct: Product
              subsets linkedCart.(Selection::selectedProduct); }
```

- Cross features in association namespace.

Redundant Cross Multiplicities (Proposal)



Cross features

Model (M1)

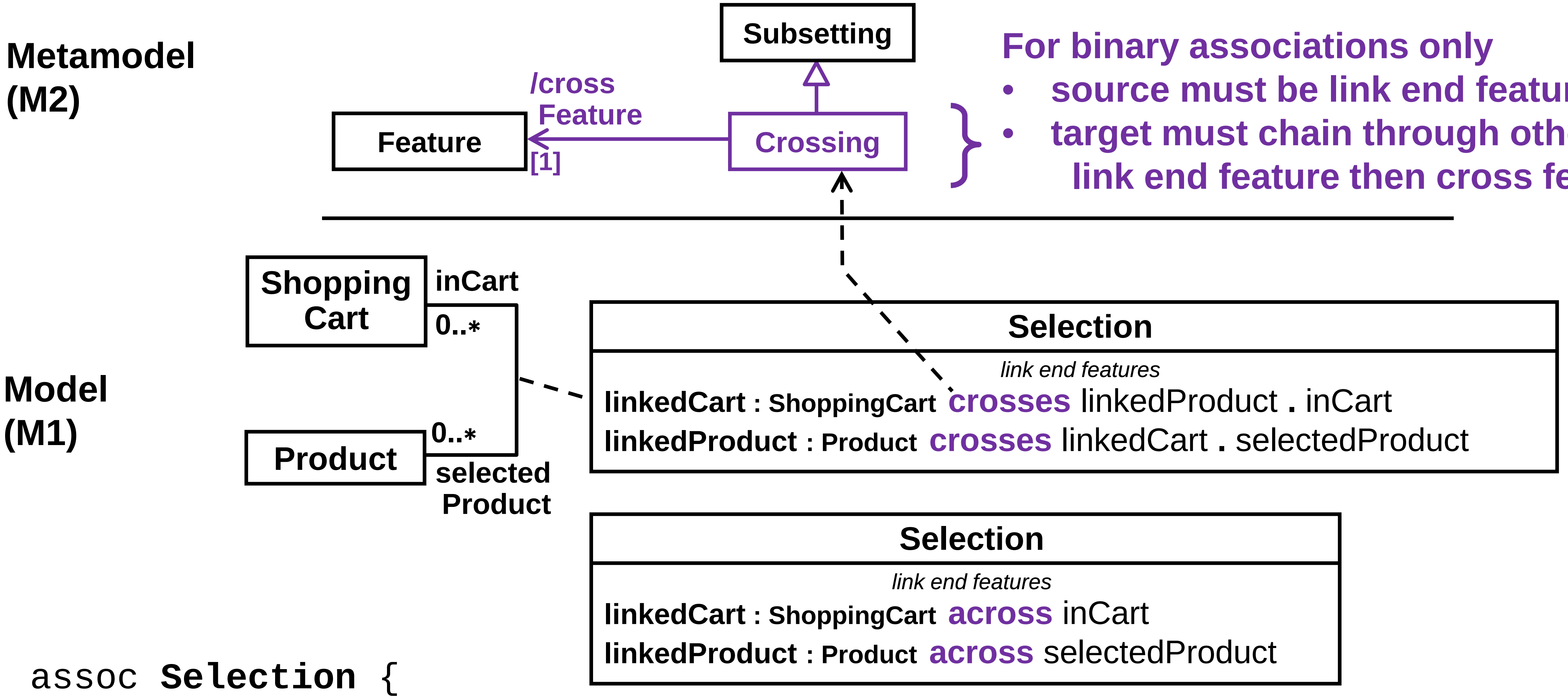
```
classifer ShoppingCart {
  feature selectedProduct: Product [1..*]; }
classifer Product {
  feature inCart: ShoppingCart [0..1]; }
assoc Selection {
  linkEnd [0..1] linkedCart: ShoppingCart
                 subsets linkedProduct.inCart;
  linkEnd [1.. *] linkedProduct: Product
               subsets linkedCart.selectedProduct; }
```

~~⊕ feature inCart: ShoppingCart [0..1] featured by Product;~~

~~⊕ feature selectedProduct: Product [1..*] featured by ShoppingCart;~~

- § **Move cross features to associated classifiers**
  - – **As UML/SysML tools currently do with “association-owned ends”.**

Assoc, Identify Cross Features (Proposal)


Metamodel
(M2)
Subsetting
/cross
Feature
[1]
Feature
Crossing
Model
(M1)
Shopping
Cart
inCart
0..*
0..*
Product
selected
Product
Selection
link end features
linkedCart : ShoppingCart crosses linkedProduct . inCart
linkedProduct : Product crosses linkedCart . selectedProduct
Selection
link end features
linkedCart : ShoppingCart across inCart
linkedProduct : Product across selectedProduct

For binary associations only

- source must be link end feature
- target must chain through other link end feature then cross feature

```
assoc Selection {
  linkEnd [0..*] linkedCart: ShoppingCart across inCart;
  linkEnd [0..*] linkedProduct: Product across selectedProduct;}
```

Overview

- Associations
- Debt (SysML1 & SST)
  - Paid (SST February)
  - Unresolved (still paying interest)
- Proposals
- **Summary**

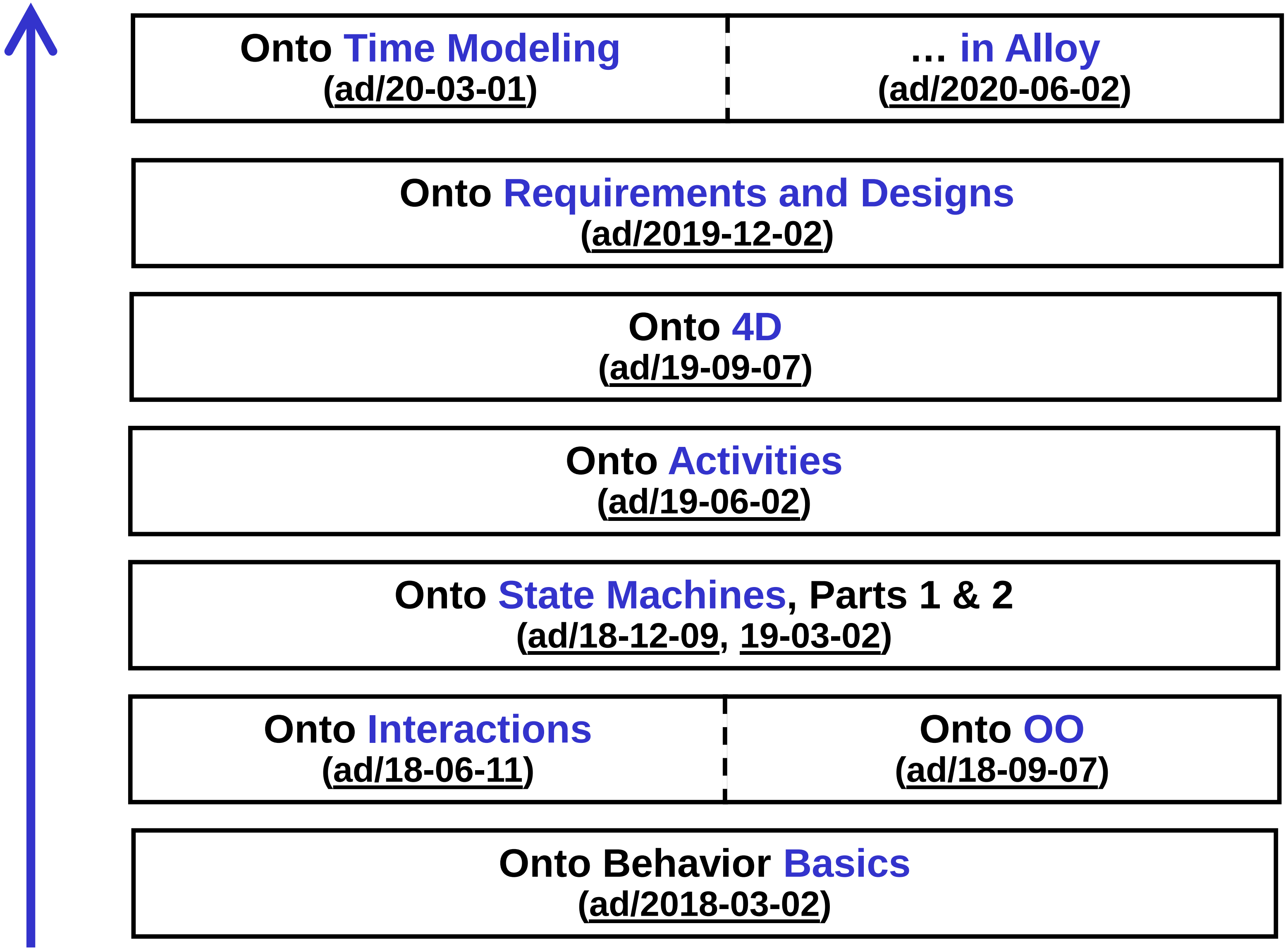
Summary

- Association (“ends”): Object oriented and relational.
  - “in” or “between” the associated classes/blocks.
- Debt (paid in SST February)
  - Inconsistency between OO and relational
- Debt, TBP
  - Misleading textual/abstract syntax (participants or ends?).
  - Two meanings for multiplicity
  - Incomplete semantics for both.
  - Cross features are not identified in standard way.

Summary, TBP Debt, Proposals

- Textual syntax is misleading (participant or cross feature?)
  - Better participant feature keyword and cross multiplicity position.
  - Leaves room for participant multiplicity inherited from library.
- Two meanings for feature multiplicity
  - Abstract syntax: Addition for cross multiplicity.
  - Participant feature multiplicity reverts to usual meaning.
- Incomplete semantics for cross features
  - Cross features owned by association.
  - Prevents redundant cross multiplicities.

Behavior as Composite Structure



Onto Time Modeling (ad/20-03-01)

… in Alloy (ad/2020-06-02)

Onto Requirements and Designs (ad/2019-12-02)

Onto 4D (ad/19-09-07)

Onto Activities (ad/19-06-02)

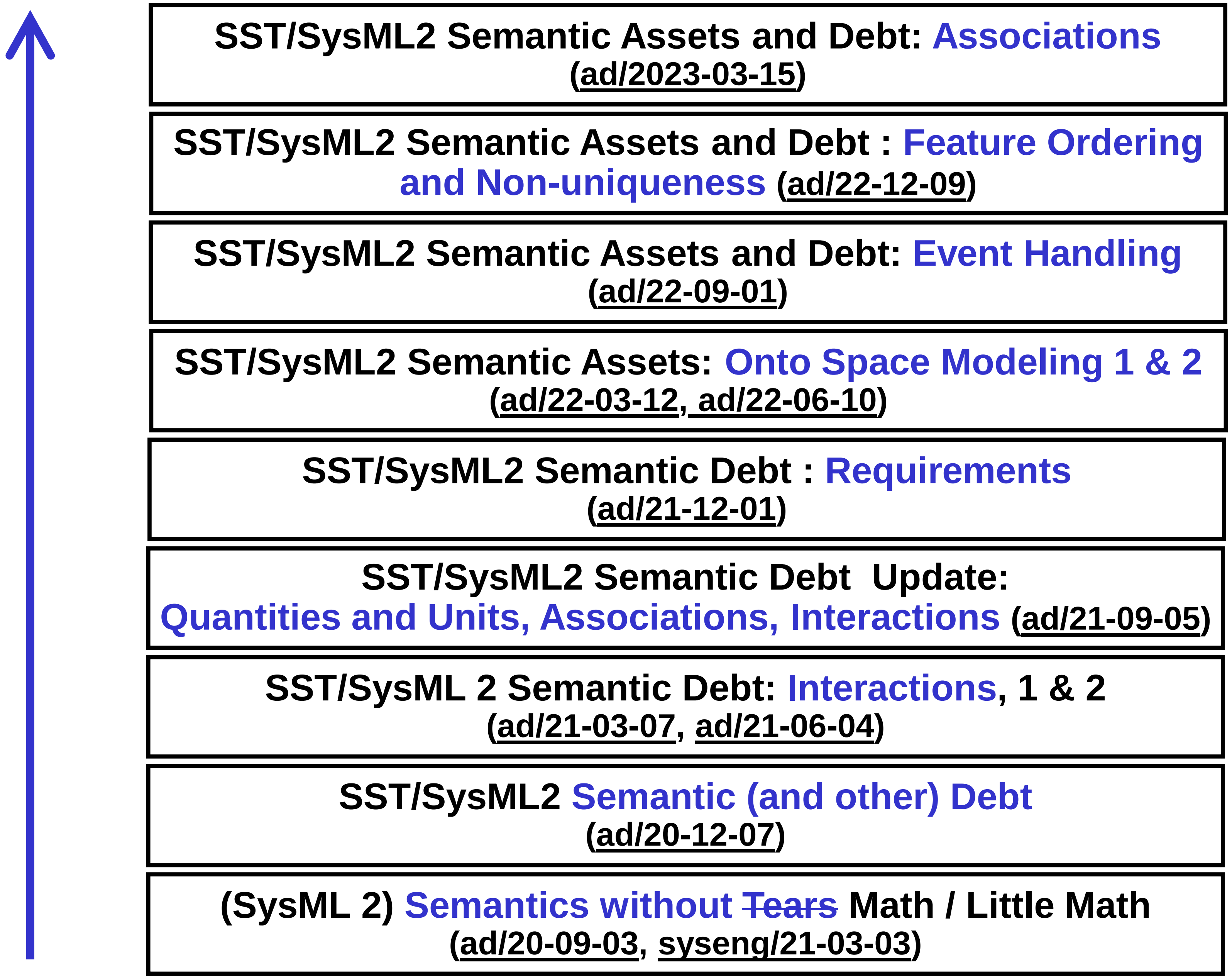
Onto State Machines, Parts 1 & 2 (ad/18-12-09, 19-03-02)

Onto Interactions (ad/18-06-11)

Onto OO (ad/18-09-07)

Onto Behavior Basics (ad/2018-03-02)

SST/SysML2 Semantic Assets and Debts



SST/SysML2 Semantic Assets and Debt: Associations (ad/2023-03-15)

SST/SysML2 Semantic Assets and Debt : Feature Ordering and Non-uniqueness (ad/22-12-09)

SST/SysML2 Semantic Assets and Debt: Event Handling (ad/22-09-01)

SST/SysML2 Semantic Assets: Onto Space Modeling 1 & 2 (ad/22-03-12, ad/22-06-10)

SST/SysML2 Semantic Debt : Requirements (ad/21-12-01)

SST/SysML2 Semantic Debt Update:
Quantities and Units, Associations, Interactions (ad/21-09-05)

SST/SysML 2 Semantic Debt: Interactions, 1 & 2 (ad/21-03-07, ad/21-06-04)

SST/SysML2 Semantic (and other) Debt (ad/20-12-07)

(SysML 2) Semantics without ~~Tears~~ Math / Little Math (ad/20-09-03, syseng/21-03-03)

Presentation Archives

- Onto(logical) Behavior Modeling Archive
  - ad/2023-03-13
- SST/SysML2 Semantic Assets and Debt Archive
  - ad/2023-03-14